



**Versão
1.0.0**

SUPERIOR TRIBUNAL DE JUSTIÇA

Secretaria de Tecnologia da Informação

Coordenadoria de Desenvolvimento

PDS – STJ Ágil

Processo Ágil de Desenvolvimento
de Software do STJ

- Guia de Referência -

Índice

1 - Introdução	1
2 - Objetivo	2
3 - Escopo	2
4 - Fases	2
4.1. Fase de Entrada	3
4.2. Fase de Pré-projeto	4
4.3. Fase de Projeto e Solução.....	4
4.4. Fase de Preparação	4
4.5. Fase de Desenvolvimento	5
4.6. Fase de Homologação.....	5
4.7. Fase de Implantação	5
5 - Papéis e Responsabilidades	6
5.1. Time Scrum	6
5.2. <i>Product Owner (PO)</i>	6
5.3. Time de Desenvolvimento	7
5.4. <i>Scrum Master (SM)</i>	8
6 - Produtos.....	9
6.1. Canvas	9
6.2. Documento de Arquitetura.....	10
6.3. <i>Roadmap</i> do Projeto.....	11
6.4. Estórias de usuário	11
6.5. Estórias técnicas.....	11
6.6. <i>Mockups</i>	11
6.7. <i>Backlog</i> do Produto	11
6.8. <i>Backlog</i> da <i>Sprint</i>	12
6.9. <i>Build</i>	12
6.10. Nota de <i>Release</i>	12
7 - Cerimônias	12
7.1. Reunião Inicial	12
7.2. Planejamento da <i>Sprint</i>	13
7.3. Reunião Diária.....	13
7.4. Revisão da <i>Sprint</i>	14
7.5. Retrospectiva da <i>Sprint</i>	14
8 - Referências	15

1 - Introdução

O Processo de Desenvolvimento de Software do STJ (PDS-STJ) é baseado no framework SCRUM e contém algumas práticas de outras metodologias ágeis como o Kanban e o Extreme Programming (XP).

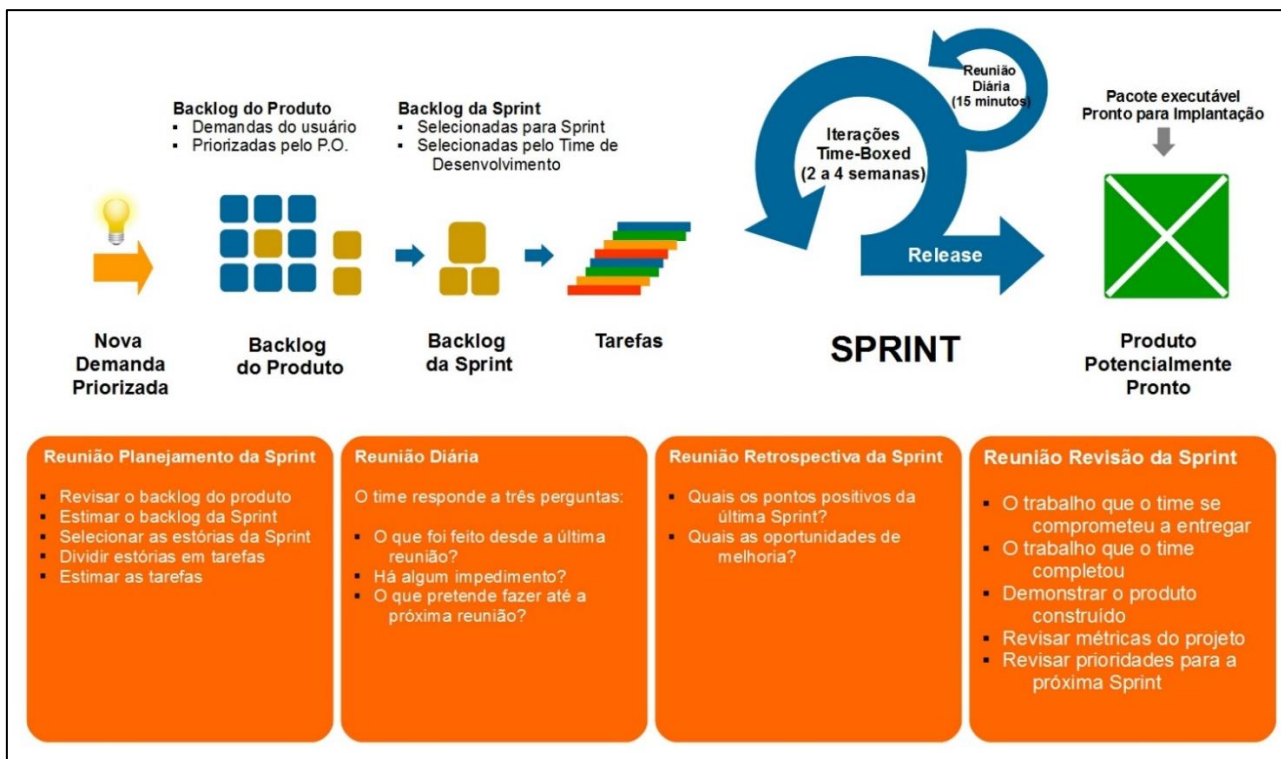


Figura 1 - Ciclo de Vida na metodologia SCRUM.

Considerando que estes modelos e práticas são de amplo conhecimento e se encontram descritos em vasta literatura, incluindo o Guia do Scrum¹ - documento oficial, trataremos no presente documento, das particularidades do processo adotado no STJ.

¹ SCHWABER, Ken. SUTHERLAN, Jeff. *Guia do Scrum®. Um guia definitivo para o Scrum: As regras do jogo.* Disponível em <http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Portuguese-Brazilian.pdf>, acesso em 01/02/2017.

2 - Objetivo

O objetivo do PDS-STJ é organizar o trabalho executado pela equipe técnica do Projeto, potencializando sua produtividade e reduzindo o retrabalho, além de melhorar a qualidade dos produtos construídos pela CDES, aumentando a satisfação das áreas demandantes e diminuindo o número de demandas corretivas recebidas.

3 -

Escopo

O escopo do PDS-STJ é o desenvolvimento interno de projetos e evolução de soluções de *software* do STJ, tanto para a área administrativa (sistemas administrativos), quanto para a área fim (sistemas judiciais). Neste documento, o termo “projeto” é utilizado tanto para projetos novos de *software*, quanto para evoluções em sistemas já existentes nestas duas áreas.

4 - Fases

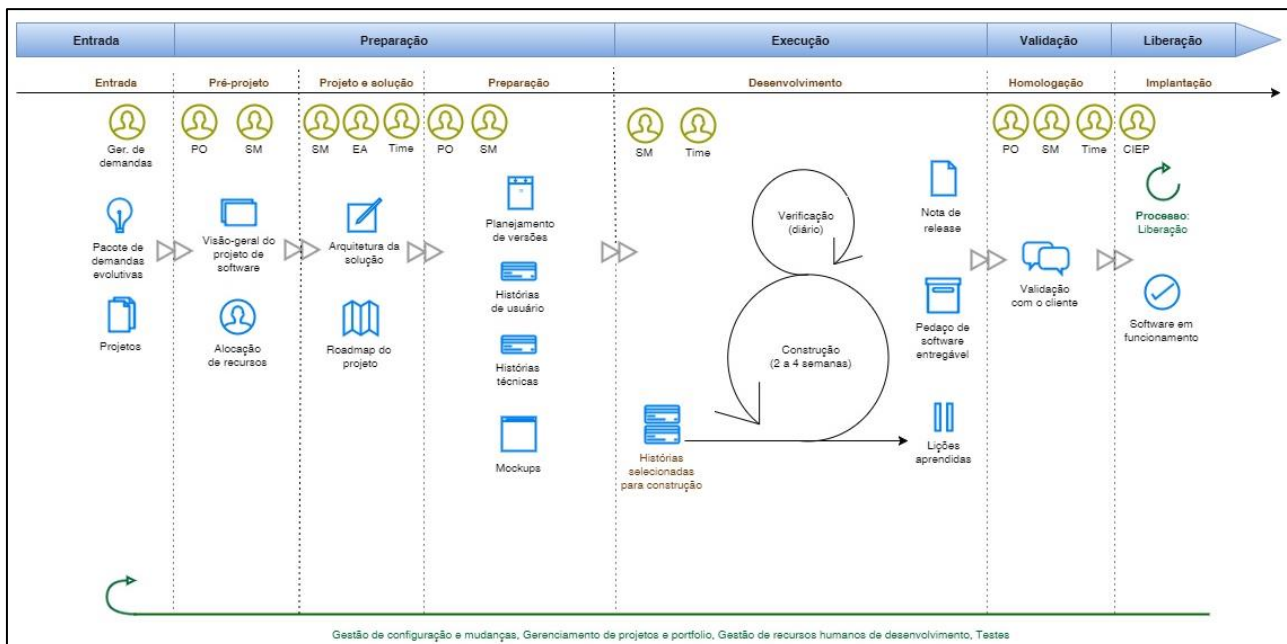
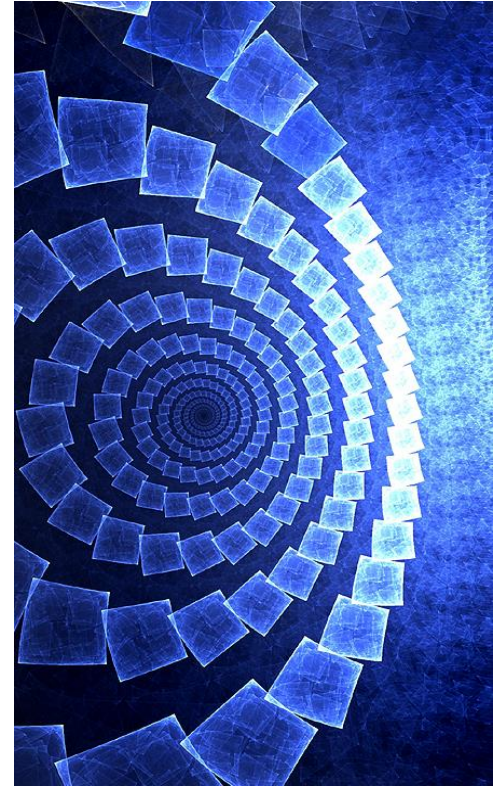


Figura 2 - Processo de desenvolvimento de soluções de *software* - CDES

A Figura 2 apresenta a visão geral do processo de desenvolvimento de *software* da CDES.

O grupo de fases definido no início da figura (composto por Entrada, Preparação, Execução, Validação e Liberação) representa a Cadeia de Valor da CDES e tem como objetivo relatar o andamento dos projetos para os interessados.

O segundo grupo (composto por Entrada, Pré-projeto, Projeto e Solução, Preparação, Desenvolvimento, Homologação e Implantação) são as fases que estão intimamente ligadas ao processo de desenvolvimento de *software*, e tem como finalidade guiar a equipe de desenvolvimento na construção do (s) produto (s) de *software* de cada projeto.



É importante ressaltar que as técnicas, cerimônias e ciclo de vida presentes no *framework* SCRUM são aplicados exclusivamente nas fases de preparação e desenvolvimento.

Este processo é um complemento da política de gestão de demandas da STI.

4.1. Fase de Entrada

Objetivo: Tomar conhecimento e validar as solicitações de desenvolvimento de soluções de *software*.

Tarefas:

1. Registrar as demandas no portfólio de demandas e projetos.
2. Planejar as demandas evolutivas e corretivas em pacotes de melhorias do *software*.

4.2. Fase de Pré-projeto

Objetivo: Analisar as solicitações e elaborar a visão geral do projeto.

Tarefas:

1. Elaborar a visão geral do projeto - modelo Canvas.

4.3. Fase de Projeto e Solução

Objetivo: Identificar as principais características técnicas e arquiteturais do produto, suas dependências e relacionamentos com outras soluções e as configurações de ambiente necessárias.

Tarefas:

1. Realizar análise de impacto na arquitetura da solução;
2. Elaborar o documento de arquitetura da solução;
3. Elaborar o roadmap do projeto.

4.4. Fase de Preparação

Objetivo: Definir o time que trabalhará no desenvolvimento, atualizar e priorizar os itens do *backlog* do produto, detalhar os requisitos dos itens prioritários e planejar as sprints de construção.

Tarefas:

1. Atualizar o *backlog* do produto;
2. Elaborar as histórias de usuários e/ou histórias técnicas;
3. Elaborar *mockups* de telas;
4. Elaborar modelo de dados;
5. Planejar as versões e *sprints*;
6. Alocar recursos;
7. Preparar o ambiente de desenvolvimento e controle de versão.

4.5. Fase de Desenvolvimento

Objetivo: Construir a solução de *software*. Nesta fase ocorrem as *sprints* de construção do produto, conforme definido no Guia do Scrum²

Tarefas:

1. Desenvolver as histórias selecionadas para a *sprint*;
2. Gerar o *build* entregável;
3. Elaborar o documento de lições aprendidas da *sprint*;
4. Elaborar nota de *release*.

4.6. Fase de Homologação

Objetivo: Assegurar que o pedaço de *software* entregável está pronto e válido conforme esperado pelo cliente.

Tarefas:

1. Publicar o *build* em ambiente de homologação;
2. Validar a solução com o cliente.

4.7. Fase de Implantação

Objetivo: Implantar a solução de *software*, garantir a comunicação aos interessados e encerrar a solicitação.

Tarefas:

1. Solicitar a implantação do *build* em ambiente de produção;
2. Realizar as comunicações aos interessados;
3. Realizar a pesquisa de satisfação.

² SCHWABER, Ken. SUTHERLAN, Jeff. *Guia do Scrum*®. *Um guia definitivo para o Scrum: As regras do jogo*. Disponível em <http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Portuguese-Brazilian.pdf>, acesso em 01/02/2017.

5 - Papéis e Responsabilidades

Os esforços de desenvolvimento utilizando Scrum consistem em uma ou mais equipes Scrum, cada uma composta basicamente de três papéis. A seguir, apresentamos a definição tanto da equipe (time) quanto dos papéis representados pelos seus membros:

5.1. Time Scrum

O Time Scrum é composto pelo *Product Owner*, o Time de Desenvolvimento e o *Scrum Master*. Times Scrum são auto organizáveis e multifuncionais. Times auto organizáveis escolhem qual a melhor forma para completarem seu trabalho, em vez de serem dirigidos por outros fora do time.

Times multifuncionais possuem todas as competências necessárias para completar o trabalho sem depender de outros que não fazem parte da equipe. O modelo de time no Scrum é projetado para aperfeiçoar a flexibilidade, criatividade e produtividade.

Times Scrum entregam produtos de forma iterativa e incremental, maximizando as oportunidades de realimentação. Entregas incrementais de produto “Pronto” garantem que uma versão potencialmente funcional do produto do trabalho esteja sempre disponível.

5.2. Product Owner (PO)

O *Product Owner*, ou dono do produto, é o responsável por maximizar o valor do produto e do trabalho do Time de Desenvolvimento. Ele é o único responsável por gerenciar o *Backlog* do Produto. Ele quem certifica que o time está mirando o alvo certo, apontando o time para a direção correta a fim de alcançar o objetivo da forma mais eficiente possível.

As principais responsabilidades do *Product Owner* são:

1. Expressar claramente os itens do *backlog* do Produto;
2. Ordenar os itens de *Backlog* do Produto para alcançar melhor as metas e missões

3. Garantir o valor do trabalho realizado pelo Time de Desenvolvimento
4. Garantir que o *Backlog* do Produto seja visível, transparente, claro para todos, e mostrar o que o Time Scrum vai trabalhar a seguir e
5. Garantir que o Time de Desenvolvimento entenda os itens do *Backlog* do Produto no nível necessário.

5.3. Time de Desenvolvimento

A equipe de desenvolvimento do Scrum é formada por todos aqueles que trabalham na construção das funcionalidades do *software* de fato, que produzirão o código-fonte que formará o produto vislumbrado pelo cliente. São eles que realizam o trabalho de entregar uma versão usável que potencialmente incrementa o produto “Pronto” ao final de cada *Sprint*. Somente integrantes do Time de Desenvolvimento criam incrementos.

Os Times de Desenvolvimento são estruturados e autorizados pela organização para organizar e gerenciar seu próprio trabalho. A sinergia resultante aperfeiçoa a eficiência e a eficácia do Time de Desenvolvimento como um todo.

Os Times de Desenvolvimento têm as seguintes características:

1. Eles são auto organizados. Ninguém (nem mesmo o *Scrum Master*) diz ao Time de Desenvolvimento como transformar o *Backlog* do Produto em incrementos de funcionalidades potencialmente utilizáveis;
2. Times de Desenvolvimento são multifuncionais, possuindo todas as habilidades necessárias, enquanto equipe, para criar o incremento do Produto.
3. O Scrum não reconhece títulos para os integrantes do Time de Desenvolvimento que não seja o Desenvolvedor,

independentemente do trabalho que está sendo realizado pela pessoa; não há exceções para esta regra.

4. Individualmente os integrantes do Time de Desenvolvimento podem ter habilidades especializadas e área de especialização, mas a responsabilidade pertence ao Time de Desenvolvimento como um todo; e,
5. Times de Desenvolvimento não contém sub-times dedicados a domínios específicos de conhecimento, tais como teste ou análise de negócios.

5.4. Scrum Master (SM)

O *Scrum Master* é responsável por garantir que o SCRUM seja entendido e aplicado, e que o time esteja aderindo aos valores, às práticas e às regras deste framework. Ele é uma espécie de técnico do time, ensinando e liderando o time de desenvolvimento na criação de produtos de *software* com valor para o usuário.

As principais responsabilidades do *Scrum Master* são:

1. Facilitar os eventos Scrum conforme exigidos ou necessários;
2. Ensinar o Time de Desenvolvimento no autogerenciamento, na interdisciplinaridade e na criação de produtos de alto valor;
3. Remover os impedimentos para o progresso do Time de Desenvolvimento;
4. Liderar e treinar a CDES na adoção do Scrum;
5. Planejar implementações do Scrum dentro da CDES;
6. Ajudar as partes interessadas a compreender e tornar aplicável o Scrum e o desenvolvimento de produto empírico;
7. Causar mudanças que aumentam a produtividade do Time Scrum;

8. Trabalhar com outros *Scrum Masters* para aumentar a eficácia da aplicação do Scrum na CDES.

6 - Produtos

Dentro da metodologia Scrum, alguns documentos são essenciais para que o trabalho consiga ser realizado com sucesso e a metodologia seja aplicada seguindo essa mesma linha. Cada um deles possui uma função específica e precisa ser desenvolvido para que no final se tenha não só um projeto concluído, mas também uma documentação completa. Dentre os principais produtos utilizados na metodologia, podemos destacar:

6.1. Canvas

Objetivo: expor a visão geral do projeto de desenvolvimento de *software*, contendo:

1. As justificativas do trabalho a ser realizado;
2. Os objetivos do projeto ou pacote de evoluções;
3. Os benefícios a serem alcançados após a implementação;
4. O produto a ser entregue para suprir a necessidade;
5. Os requisitos do produto, sem detalhamento, apenas uma ideia das funcionalidades;
6. O grupo de entregas, que são partes do produto final e estão relacionados com os requisitos;
7. A linha do tempo, que deve indicar uma sequência de execução o grupo de entregas, em ordem de prioridade definida pelo PO;
8. Os stakeholders e fatores externos do projeto ou pacote de melhorias;
9. A equipe que desempenhará os papéis Scrum no projeto;
10. As premissas do projeto, que devem ser avaliadas;

11. Os riscos do projeto, que devem ser tratados;
12. As restrições do projeto, que devem ser levadas em conta para estipulação de prazos.

Responsáveis: *Scrum Master, Product Owner.*

Modelo:

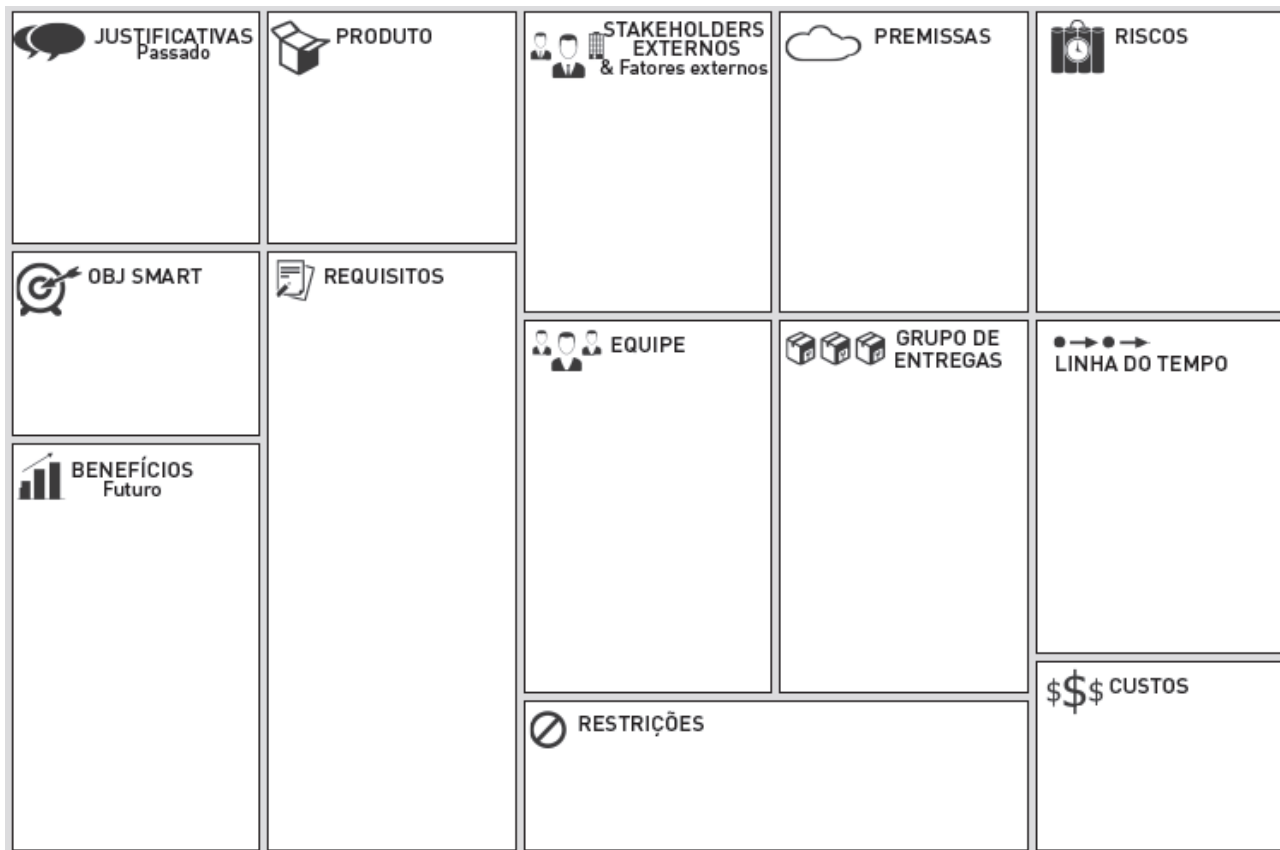


Figura 3 - Modelo de Canvas de Projeto

6.2. Documento de Arquitetura

Objetivo: documentar, de forma visual, objetiva e clara, a arquitetura em que o projeto será construído.

Responsável: Time de desenvolvimento, Equipe de Arquitetura.

6.3. Roadmap do Projeto

Objetivo: detalhar a linha do tempo elaborada no Canvas, estipulando prazo para, no máximo, duas entregas.

Responsáveis: *Scrum Master, Product Owner*, Time de Desenvolvimento.

6.4. Estórias de usuário

Objetivo: detalhar os requisitos para cada entrega, levantando quem são os atores, a funcionalidade a ser executada e o motivo negocial pelo qual esta deve ser implementada.

Responsável: *Product Owner*.

6.5. Estórias técnicas

Objetivo: Detalhar as necessidades técnicas levantadas pelo time de desenvolvimento que fazem parte do produto gerado a partir de uma estória de usuário, mas que são transparentes para ele.

Responsável: Time de Desenvolvimento.

6.6. Mockups

Objetivo: apresentar graficamente (em forma de protótipo não funcional) as funcionalidades presentes nas estórias de usuário, a fim de facilitar o entendimento dos requisitos por todas as partes envolvidas e a construção dos critérios de aceitação.

Responsável: Time de Desenvolvimento.

6.7. Backlog do Produto

Objetivo: agrupar todas as estórias de usuário e técnicas que fazem parte do produto. Priorizar as estórias em ordem de importância do projeto.

Responsável: *Product Owner*.

6.8. **Backlog da Sprint**

Objetivo: reunir todas as histórias de usuário e técnicas selecionadas pelo PO para a implementação na *sprint*.

Responsável: *Product Owner*.

6.9. **Build**

Objetivo: disponibilizar ao usuário parte do produto a fim de entregar valor ao usuário, suprimindo algumas das necessidades levantadas.

Responsável: Time de Desenvolvimento.

6.10. **Nota de Release**

Objetivo: documentar as funcionalidades implementadas no *build* entregue, a fim de manter a rastreabilidade entre a versão implantada e os requisitos.

Responsável: Time de Desenvolvimento.

7 - Cerimônias

As cerimônias no Scrum são eventos que acontecem dentro de um ciclo de desenvolvimento utilizando esta metodologia. Existem cinco tipos de cerimônias:

7.1. **Reunião Inicial**

Quando uma demanda de projeto ou um pacote de demandas evolutivas é priorizado pela Assessoria de Planejamento da STI, é necessário realizar uma reunião inicial com representantes que conheçam a arquitetura em que a solução será construída e um *Scrum Master* para que sejam definidos objetivos, recursos, restrições e estipulados prazos para o projeto.

A fim de discutir a arquitetura, um integrante da SCORP deve participar da reunião, uma vez que esta Seção possui a responsabilidade de manter a arquitetura e os padrões de *software* da CDES. Caso seja uma evolução de algum módulo já existente, além do integrante da SCORP, deve participar um

especialista no módulo a ser evoluído, de modo a avaliar as soluções possíveis para a necessidade apresentada. Caso seja necessário um aval de viabilidade de infraestrutura, é de responsabilidade dos envolvidos com a arquitetura contatar servidores da Coordenadoria de Infraestrutura para esta avaliação.

7.2. Planejamento da *Sprint*

Sempre antes de iniciar uma *sprint*, deve-se realizar o seu planejamento em uma reunião, a qual tem seu tempo máximo estipulado em 8 horas, dividido em duas partes de 4 horas cada. É realizada para que o time defina, basicamente “o que será feito” e “como será feito”.

A primeira etapa tem o objetivo de atualizar o *Backlog* do Produto de acordo com a visão de prioridade do cliente e com o que já fora implementado.

Na segunda etapa prepara-se o *Backlog* da *Sprint*, que deve conter histórias de usuário selecionadas do *Backlog* do Produto de acordo com a priorização do *Product Owner*, que normalmente escolhe aqueles requisitos que ele enxerga mais valor para o negócio. Essa priorização, no início do projeto, pode sofrer interferência do *Scrum Master*, se este que enxergar a necessidade de implementar histórias que exigem a construção da arquitetura completa do *software*, ou seja, que precise passar por todas as camadas.

Após a escolha das histórias, é feita uma quebra delas em tarefas que o time consiga estipular o tempo necessário para a completude de cada uma, reduzindo, assim, distorções nas estimativas de tempo para a execução delas. Para cada tarefa, um ou mais responsáveis pela sua execução darão o *feedback* do seu andamento nas reuniões diárias.

7.3. Reunião Diária

Durante a *sprint*, diariamente acontece uma reunião de tempo definido e curto (15 minutos), executada preferivelmente em pé. Este evento tem como objetivo principal que cada membro do Time explique brevemente:

1. O que realizou desde a última reunião?
2. O que irá realizar até a próxima reunião?
3. Quais obstáculos ou impedimentos encontrou ao tentar completar essas atividades?

7.4. Revisão da *Sprint*

O fim de uma *sprint* é marcado por uma reunião, chamada de revisão da *sprint*, em que é feita a verificação e validação das funcionalidades construídas para liberação de um pacote ao cliente. Neste evento todos participam: *Product Owner*, *Scrum Master*, time e *stakeholders*. Essa reunião não pode ultrapassar quatro horas e tem como objetivo apresentar ao *Product Owner* e aos *stakeholders* as funcionalidades que foram construídas de acordo com o planejamento da *sprint*, mostrando quais histórias de usuário do *Backlog* do Produto estão implementadas com base nos critérios de aceitação de cada história.

Após a apresentação, os *stakeholders*, juntamente com o *Product Owner*, devem aprovar ou rejeitar os trabalhos completados, bem como expor suas impressões sobre o produto e informar, caso haja, mudanças desejadas e classificá-las em níveis de prioridade. Caso vislumbrem alguma nova funcionalidade que não faz parte do *Backlog* do Produto, eles podem pedir a inclusão da mesma, indicando a prioridade.

Com essa reunião encerrada, a *sprint* é fechada. Se houver alguma história que deveria ter sido implementada e, por algum motivo, não foi totalmente construída, esta deve ser incluída na pauta da reunião de planejamento da próxima *sprint*.

Caso ainda haja itens de *backlog* no projeto, o ciclo Scrum reinicia e uma nova iteração deve ser iniciada, com uma nova reunião de planejamento da *Sprint*. Este ciclo ocorre até que todos os itens que compõem o *Backlog* do produto sejam construídos e o pacote de trabalho do projeto seja finalizado.

7.5. Retrospectiva da *Sprint*

Após o encerramento de uma *sprint*, acontece uma reunião chamada retrospectiva da *sprint*, que tem o objetivo de reunir o Time, o *ScrumMaster* e, o *Product Owner*, para que eles discutam o que ocorreu bem durante a *sprint* que se encerrou e o que pode ser melhorado para a próxima.

8 - Referências

SCHWABER, Ken. SUTHERLAN, Jeff. *Guia do Scrum®: Um guia definitivo para o Scrum: As regras do jogo*. Disponível em <http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Portuguese-Brazilian.pdf>, acesso em 01/02/2017.

SUTHERLAND, Jeff. *Scrum. A arte de fazer o dobro na metade do tempo*. Editora Leya: out/2016.

CRUZ, Fábio. *Scrum e Agile em Projetos - Guia Completo: Conquiste sua certificação e aprenda a usar métodos ágeis no seu dia a dia*. Editora Brasport: Fev/2015.